13th July 2021

Lab 7 – DFT for signal analysis

Objectives: In this lab we will use DFT to process signals

7.1 DFT for frequency analysis of CT signals

Reading prerequisite: Section 7.4 of the Proakis textbook

Consider the continuous-time signal $p(t) = \cos(\omega_0 t) = \cos(2\pi f_0 t)$. Let $p(t)$ be sampled at frequency $f_s = \frac{1}{T_s}$ which is greater than Nyquist rate giving $p[n] = p(nT_s)$. We want to estimate frequency f_0 of $p(t)$ from finite number of its samples.

(a) [CTFT] What is the Fourier transform $P(\omega)$ of $p(t)$?

(b) [DTFT] What is the Fourier transform $P(e^{j\omega})$ of $p[n]$?

(c) From (a) and (b), what is the relation between the location of the impulses?

Consider a finite number of samples of $p[n]$ which can be obtained by the windowing operation $x[n] = p[n] \times w[n]$ where $w[n]$ is the rectangular window function which is 1 for $0 \leq n \leq L-1$ and 0 otherwise.

(d) [DTFT] What will be the Fourier transform $X(e^{j\omega})$ of $x[n]$?

Hint: multiplication property of DTFT. What is the effect of the window on spectrum? (focus on magnitude spectrum for simplicity)

Any practical signal processing will involve a window function $w[n]$ of some kind to get finite length sequences. We now consider $x[n]$ as an L-length sequence and study it using **DFT**.

Write a **matlab script** for the following tasks.

(e) Let $f_0 = 12\ Hz$ and $f_s = 64\ Hz$. Generate samples $x[n]$ for **L = 16**. Compute its DFT $X[k]$ of length-N using the `fft` command and plot magnitude of DFT for N = mL where m = {1,2,4,8}. Plot all of them in a single 2x2 figure. Are your plots consistent with the answer in part (d)? (read about spectral leakage from Proakis 7.4).

Note that `fft` command automatically performs required zero-padding to compute length-N DFT (if signal length L < N). For m = {1,2} use stem, for m = {4,8} use plot.

(f) Now repeat above for changing L = {16, 32, 64, 128} and fixed N = 8L. Plot magnitude spectrum in a single 2x2 figure. Comment on spectrum shape as L changes? What can you conclude about the length of the signal (L) and the frequency resolution?

(g) Repeat (e) when $f_0 = 11\ Hz$.

(h) Repeat (e) when $w[n]$ is an L-length Hanning window. Use the matlab command `hann` to get this window. Comment on changes in main-lobe width and spectral leakage compared to the figure in part (e).

(i) From the plots of DFT magnitude spectrum, how would you estimate f_0 given f_s? Use this method to estimate f_0 in part (e), (g) and (h). Does N affect your answers?

(j) Load one of the audio files shared in the repository, you should load the file number given by team_number (modulo 12). Use the analysis above to find the three strongest frequencies (in Hz) present in the audio signal. You are encouraged to process other files as well.

7.2 Direct and DFT based convolutions

We can compute time-domain convolutions using various methods. In this **matlab script** we will compute linear convolution and circular convolution of a pair of signals using two methods for each.

(a) Generate two finite length sequences as follows. The sequence $x_1[n]$ is a random Gaussian sequence of length 10 and $x_2[n]$ is first 10 samples of the signal $\delta[n-3]$ starting from n = 0.

(b) Perform linear and circular convolutions of $x_1[n]$ and $x_2[n]$ directly using the commands `cconv` and `conv`, respectively (read up MATLAB documentation of these commands). Make sure each result is of the expected length.

(c) Perform linear and circular convolutions of $x_1[n]$ and $x_2[n]$ using the DFT method. You must perform these convolutions using DFT and inverse DFT of the appropriate signals. Use the matlab commands `fft` and `ifft` to compute the DFT and inverse DFT of the signals. Make sure each result is of the expected length.

(d) Plot the four outputs in a single 2x2 figure and verify that the two methods (direct and DFT based) give same answer for linear convolution. Similarly for circular convolution.

7.3 DFT of some signals

In this task we will compute DFT of some standard signals using the built-in `fft` command. In a **matlab script** compute DFT for various N-length sequences given as follows:

a) [ones(L,1); zeros(N-L,1)], for fixed L = 4 repeat for N = 4, 16, 64

b) $\sin(\omega_0 n)$, for $\omega_0 = 3\pi/10$ and N = 20

c) $\cos(\omega_0 n)$, for $\omega_0 = 3\pi/10$ and N = 20

d) $\sin(\omega_0 (n-1))$, $\omega_0 = 3\pi/10$ and N = 20

e) $(0.8)^n$ for N = 20

f) $(-0.8)^n$ for N = 20

In the same 3x1 figure, plot the sequence and the magnitude and phase spectrum of its DFT. Can you identify the low frequencies and high frequencies from the spectrum?

7.4 The radix-2 FFT (bonus, optional)

Write a **matlab function** `radix2fft` which takes as input an N-length vector x and returns an N-length vector X. To compute X, implement the decimation-in-time radix-2 FFT algorithm for an input vector whose length is a power of 2 i.e. assume that $N = 2^m$. Note that this function will have a recursive structure. Specifically, `radix2fft` is called within itself until it reaches the stopping criteria of N = 2. What is the DFT when N = 2? Verify that the output of your function matches with that of `fft` within numerical precision. This can be attempted after the next class.